



A Sierra Monitor Company

Driver Manual
(Supplement to the FieldServer Instruction Manual)

FS-8700-127 Hanning & Kahl
HCS-R

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after May 1, 2001

Driver Version:	1.00
Document Revision:	9

TABLE OF CONTENTS

1. HANNING & KAHL HCS-R DESCRIPTION..... 3

2. DRIVER SCOPE OF SUPPLY 4

 2.1. Supplied by FieldServer Technologies for this driver..... 4

3. HARDWARE CONNECTIONS..... 5

 3.1. Hardware Connection Tips / Hints..... 5

4. CONFIGURING THE FIELDSERVER AS A HCS-R CLIENT 6

 4.1. Data Arrays/Descriptors..... 6

 4.2. Client Side Connection Descriptions 7

 4.3. Client Side Node Descriptors 7

 4.4. Client Side Map Descriptors..... 8

 4.4.1. *FieldServer Related Map Descriptor Parameters* 8

 4.4.2. *Driver Related Map Descriptor Parameters*..... 8

 4.4.3. *Timing Parameters* 8

 4.5. Map Descriptor Example: Read Data 9

 4.6. Map Descriptor Example: Read Data and Keep History 10

5. CONFIGURING THE FIELDSERVER AS A HCS-R SERVER 11

 5.1. Server Side Connection Descriptors 11

 5.2. Server Side Node Descriptors..... 12

 5.3. Server Side Map Descriptors 13

 5.3.1. *FieldServer Specific Map Descriptor Parameters*..... 13

 5.3.2. *Driver Specific Map Descriptor Parameters* 13

 5.3.3. *Map Descriptor Example. - SMD_HCSR_01 Map Descriptor to Serve HCS-R clients.* 14

APPENDIX A. ADVANCED TOPICS 15

 Appendix A.1. 1. How Client Stores Data:..... 15

 Appendix A.1.1. *Data_Type is Ana_Input or not specified*..... 15

 Appendix A.1.2. *Data_Type is Digi_Input*..... 15

 Appendix A.2. 2. How Server expects Data:..... 16

 Appendix A.2.1. *Data_Type is Ana_Input or not specified*..... 16

 Appendix A.2.2. *Data_Type is Digi_Input*..... 16

1. Hanning & Kahl HCS-R Description

The Hanning & Kahl HCS-R driver allows the transfer of data from devices over RS-232 HCS-R protocol. Data is read from HCS-R and current data stored in a specified memory area. The driver can store the data with its loop data status distinct memory areas depending on whether the data is on loop or from loop. The driver is based on section 10.5 of Hanning & Kahl - part number 40801121. The FieldServer can act as a Client or Server.

Max Nodes Supported

FieldServer Mode	Nodes	Comments
Client	1	Only 1 virtual client per port allowed on multidrop systems,
Server	1	This is a protocol limit, maximum devices can be polled by Client

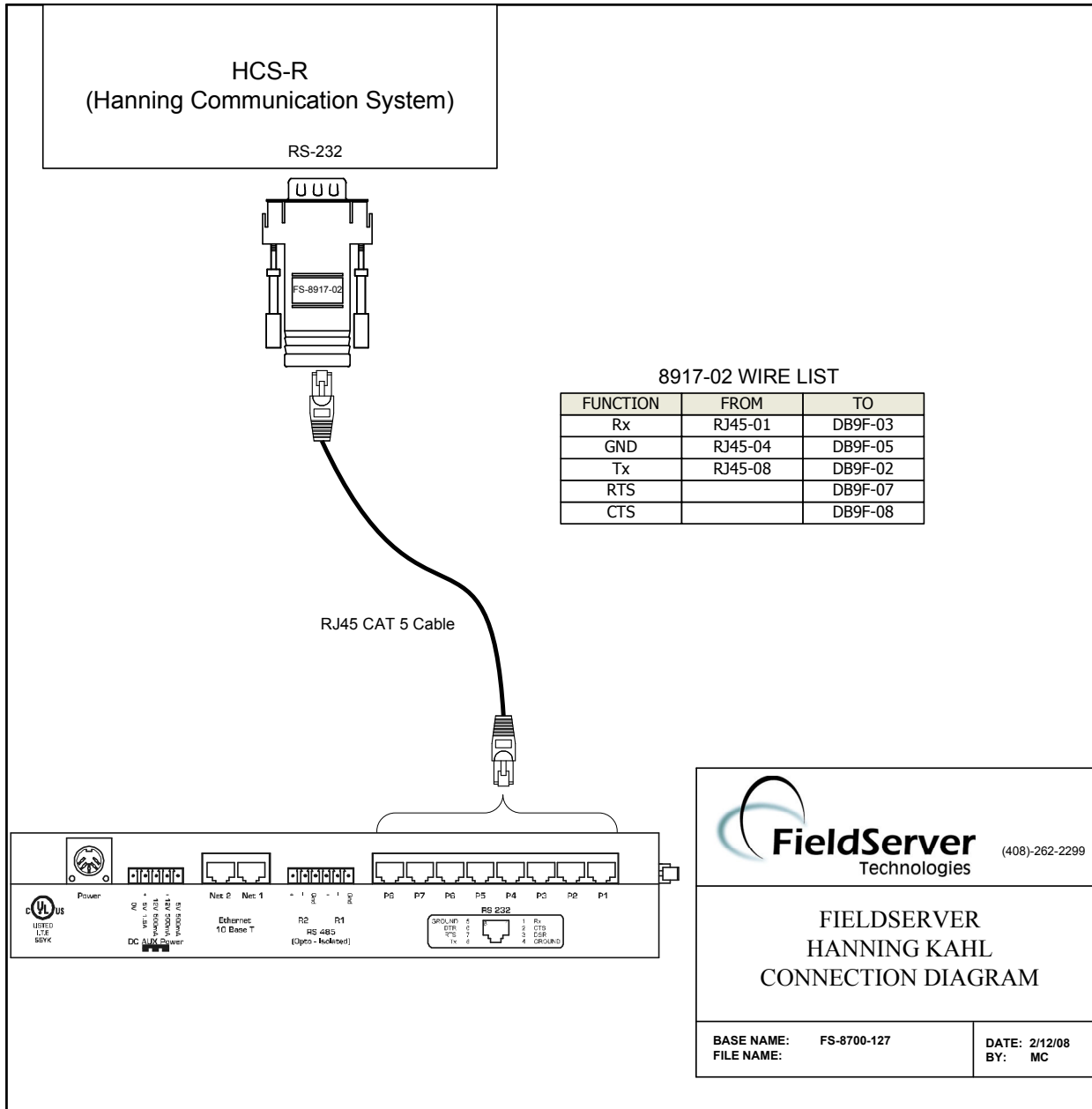
2. Driver Scope of Supply

2.1. Supplied by FieldServer Technologies for this driver

FieldServer Technologies PART #	Description
FS-8915-10	UTP cable (7 foot) for RS-232 use
FS-8700-127	Driver Manual.

3. Hardware Connections

The FieldServer is connected to the Hanning & Kahl HCS-R as shown in connection drawing. Configure the HCS-R according to manufacturer's instructions



3.1. Hardware Connection Tips / Hints

1. Loop back RTS CTS if it is required by the other device.

4. Configuring the FieldServer as a HCS-R Client

For a detailed discussion on FieldServer configuration, please refer to the FieldServer Configuration Manual. The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer (See “.csv” sample files provided with the FieldServer).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a HCS-R Server.

4.1. Data Arrays/Descriptors

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for HCS-R communications, the driver independent FieldServer buffers need to be declared in the “Data Arrays” section, the destination device addresses need to be declared in the “Client Side Nodes” section, and the data required from the servers needs to be mapped in the “Client Side Map Descriptors” section. Details on how to do this can be found below.

Note that in the tables, * indicates an optional parameter, with the bold legal value being the default.

Section Title			
Data Arrays	Column Title	Function	Legal Values
	Data_Array_Name	Provide name for Data Array	Up to 15 alphanumeric characters
	Data_Array_Format	Provide data format. Each Data Array can only take on one format.	Float, Bit, UInt16, SInt16, Byte,
	Data_Array_Length	Number of Data Objects. Must be larger than the data storage area required by the Map Descriptors for the data being placed in this array.	1-10,000

Example

// Data Arrays		
Data Arrays		
Data_Array_Name,	Data_Array_Format,	Data_Array_Length
DA_CURRENT,	UInt16,	22
DI_FromLoop01,	Bit,	22
DI_OnLoop01,	Bit,	22
DI_FromLoop02,	Bit,	22
DI_OnLoop02,	Bit,	22
DI_FromLoop03,	Bit,	22
DI_OnLoop03,	Bit,	22
DI_FromLoop014	Bit,	22
DI_OnLoop04,	Bit,	22

4.2. Client Side Connection Descriptions

Section Title		
Connections		
Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer	P1-P8 ¹
Protocol	Specify protocol used	HCS-R HCSR H&K HCS-R
Baud*	Specify baud rate	9600 (Vendor limitation)
Parity*	Specify parity	None (Vendor limitation)
Data_Bits*	Specify data bits	8 (Vendor limitation)
Stop_Bits*	Specify stop bits	1 (Vendor limitation)
Handshaking*	Specify hardware handshaking	None (Vendor limitation)
Poll_Delay*	Time between internal polls	0-32000 seconds, 0.05 seconds

Example

// Client Side Connections					
Connections					
Port,	Protocol,	Baud,	Parity,	Handshaking,	Poll_Delay
P8,	HCS-R,	9600,	None,	None,	0.100s

4.3. Client Side Node Descriptors

Section Title		
Nodes		
Column Title	Function	Legal Values
Node_Name	Provide name for Node	Up to 32 alphanumeric characters
Node_ID	HCS-R station address of physical server Node	240 (Protocol limit)
Protocol	Specify protocol used	HCS-R HCSR H&K HCS-R
Connection	Specify which port the device is connected to the FieldServer	P1-P8 ¹

Example

// Client Side Nodes			
Nodes			
Node_Name,	Node_ID,	Protocol,	Connection
PLC 1,	240,	HCS-R,	P8

¹ Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

4.4. Client Side Map Descriptors

4.4.1. FieldServer Related Map Descriptor Parameters

Column Title	Function	Legal Values
Map_Descriptor_Name	Name of this Map Descriptor	Up to 32 alphanumeric characters
Data_Array_Name	Name of Data Array where data is to be stored in the FieldServer	One of the Data Array names from "Data Array" section above
Data_Array_Offset	Starting location in Data Array	0 to maximum specified in "Data Array" section above
Function	Function of Client Map Descriptor	RDBC, Passive_Client

4.4.2. Driver Related Map Descriptor Parameters

Column Title	Function	Legal Values
Node_Name	Name of Node to fetch data from	One of the Node Names specified in "Client Node Descriptor" above
HCSR_OnFrom	For Passive_Client Map Descriptor specify whether data From Loop or On Loop should be stored.	On, From, OnFrom
HCSR_Loop	For Passive_Client Map Descriptor specify loop number for which data should be stored.	1,2,3,4
Length	Length of Map Descriptor 22 (min) for RDBC Map Descriptor: 24 if data_type is Digi_Input 21 for Passive_Client Map Descriptor. It also depends upon HCSR_Data_Stages parameter Refer to Appendix A	22 or 24, 21
Data_Type*	Specify how data will be stored. Refer to Appendix A	Ana_Input Digi_Input ²
HCSR_Data_Stages*	Specify the number of stages to keep history of data changes. 0 and 1 means no history	0,1,2,3

4.4.3. Timing Parameters

Column Title	Function	Legal Values
Scan_Interval	Rate at which data is polled	≥0.001s

² Default for Digi_Input

4.5. Map Descriptor Example: Read Data

The CMD_DI Map Descriptor polls HCSR-R Node_A (described in Client Side Node Descriptors sections) every 5 seconds. The other Map Descriptors are used to store the data for various loops. Refer to Appendix A for data storage format.

Map_Descriptors	Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Function	Node_name	Length	Scan_Interval
	CMD_DI_HCSR_01,	DA_CURRENT,	0,	RDBC,	Node_A,	22,	5
	Map_Descriptors						
	CMD_FromLoop_01,	DI_FromLoop01,	0,	Passive_Client,	Node_A,	21,	HCSR_Loop
	CMD_OnLoop_01,	DI_OnLoop01,	0,	Passive_Client,	Node_A,	21,	1
	CMD_FromLoop_02,	DI_FromLoop02,	0,	Passive_Client,	Node_A,	21,	2
	CMD_OnLoop_02,	DI_OnLoop02,	0,	Passive_Client,	Node_A,	21,	2
	CMD_FromLoop_03,	DI_FromLoop03,	0,	Passive_Client,	Node_A,	21,	3
	CMD_OnLoop_03,	DI_OnLoop03,	0,	Passive_Client,	Node_A,	21,	3
	CMD_FromLoop_4,	DI_FromLoop04,	0,	Passive_Client,	Node_A,	21,	4
	CMD_OnLoop_04,	DI_OnLoop04,,	0,	Passive_Client,	Node_A,	21,	4

How often (in seconds) to poll target device. Irrelevant if Function is Passive_Client

Maximum number of placeholders for data storage.

One of the Nodes declared in Client Node Section. This Node will be polled if Function is RDBC

RDBC will cause read poll and Passive_Client is used to store data returned by response.

Starting offset in Data Array to hold data

Name one of the Data Array defined in Data_Array section to store data

4.6. Map Descriptor Example: Read Data and Keep History

This Map Descriptor allows the storage of data history up to HCSR_Data_Stages. Data will be shifted to next Stage only if the data is changed. The driver divides the Data Array into sections corresponding to the number of HCSR_Data_Stages. Current data (stage1) will be stored in the first section and oldest (last stage) data will be stored in the last section in the Data array.

Map_Descriptors	Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	Function,	Node_name	Length	Scan_interval,	HCSR_Data_Stages
	CMD_DI_HCSR_01,	DA_CURRENT,	0,	RDBC,	Node_A,	72,	5,	3
Map_Descriptors	Data_Array_Name,	Data_Array_Offset,	Function,	Node_name	Length	HCSR_Onfrom	HCSR_Loop	HCSR_Data_Stages
	DI_FromLoop01,	0,	Passive_Client,	Node_A,	63,	From	1	3
	CMD_OnLoop_01,	0,	Passive_Client,	Node_A,	63,	On	1	3
	CMD_FromLoop_02,	0,	Passive_Client,	Node_A,	63,	From	2	3
	CMD_OnLoop_02,	0,	Passive_Client,	Node_A,	63,	On	2	3
	CMD_FromLoop_03,	0,	Passive_Client,	Node_A,	63,	From	3	3
	CMD_OnLoop_03,	0,	Passive_Client,	Node_A,	63,	On	3	3
	CMD_FromLoop_4,	0,	Passive_Client,	Node_A,	63,	From	4	3
	CMD_OnLoop_04,	0,	Passive_Client,	Node_A,	63,	On	4	3

The number of data stages (data changes) to be kept as history.

How often (in seconds) to poll target device. Irrelevant if Function is Passive_Client

Maximum number of placeholders for data storage.

One of the Nodes declared in Client Section. This Node will be polled if Function is RDBC

Starting offset in Data Array to hold data
RDBC will cause read poll and Passive_Client is used to store data returned by response.

Name one of the Data Array defined in Data_Array section to store data

5. Configuring the FieldServer as a HCS-R Server

For a detailed discussion on FieldServer configuration, please refer to the FieldServer Configuration Manual. The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer (See “.csv” sample files provided with the FieldServer).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a Hanning & Kahl HCS-R Client.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for HCS-R communications, the driver independent FieldServer buffers need to be declared in the “Data Arrays” section, the FieldServer virtual Node(s) needs to be declared in the “Server Side Nodes” section, and the data to be provided to the clients needs to be mapped in the “Server Side Map Descriptors” section. Details on how to do this can be found below.

Note that in the tables, * indicates an optional parameter, with the bold legal value being the default.

5.1. Server Side Connection Descriptors

Section Title		
Connections		
Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer	P1-P8 ³
Protocol	Specify protocol used	HCS-R HCSR H&K HCS-R
Baud*	Specify baud rate	9600 (Vendor Limitation)
Parity*	Specify parity	None (Vendor Limitation)
Data_Bits*	Specify data bits	8 (Vendor Limitation)
Stop_Bits*	Specify stop bits	1
Handshaking*	Specify hardware handshaking	None (Vendor Limitation)
Server_Hold_Timeout*	Specifies the time that the FieldServer will reserve the Server side connection while waiting for the Client side to update data in Data_Array (if necessary)	>1.0s, 2.0s

Example

// Server Side Connections				
Connections				
Port,	Protocol,	Baud,	Parity,	Handshaking
P8,	HCS-R,	9600,	None,	None

³ Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

5.2. Server Side Node Descriptors

Section Title		
Nodes		
Column Title	Function	Legal Values
Node_Name	Provide name for Node	Up to 32 alphanumeric characters
Node_ID	Station address of physical server Node	240
Protocol	Specify protocol used	HCS-R HCSR H&K HCS-R
Server_Hold_Timeout*	Specifies the time that the FieldServer will reserve the Server side connection while waiting for the Client side to update data in Data_Array (if necessary)	>1.0s, 2.0s

Example

```
// Server Side Nodes
Nodes
Node_Name,           Node_ID,           Protocol
HCSR_01,             240,              HCS-R
```

5.3. Server Side Map Descriptors

5.3.1. FieldServer Specific Map Descriptor Parameters

Column Title	Function	Legal Values
Map_Descriptor_Name	Name of this Map Descriptor	Up to 32 alphanumeric characters
Data_Array_Name	Name of Data Array where data is to be stored in the FieldServer	One of the Data Array names from "Data Array" section above
Data_Array_Offset	Starting location in Data Array	0 to maximum specified in "Data Array" section above
Function	Function of Server Map Descriptor	Server
Server_Hold_Timeout*	Specifies the time that the FieldServer will reserve the Server side connection while waiting for the Client side to update data in Data_Array (if necessary)	>1.0s, 2.0s

5.3.2. Driver Specific Map Descriptor Parameters

Column Title	Function	Legal Values
Node_Name	Name of Node to fetch data from	One of the Node names specified in "Client Node Descriptor" above
Length	Length of Map Descriptor	22
Data_Type*	Specify format of the data to serve. Refer to Appendix A	Digi_Input Ana_Input

5.3.3. Map Descriptor Example. - SMD_HCSR_01 Map Descriptor to Serve HCS-R clients.

```
// Server Side Map Descriptors
Map_Descriptors
Map_Descriptor_Name, Data_Array_Name, Data_Array_Offset, Function, Node_Name, Length, Scada_Hold_Timeout
SMD_HCSR_01, DA_HCSR, 0, Server, HCSR_01, 22, 5
```

One of the Data Array name described in Data_Array section. This Data Array holds the data to be served to HCS-R client.

The starting location of data to serve within the Data Array

One of the Nodes described in Server Side Node Descriptors section

The number of items within the Data Array which will be used to prepare the response to the HCS-R client. See Appendix A for more detail.

Appendix A. Advanced Topics

Appendix A.1. 1. How Client Stores Data:

The HCS will be polled by an RDBC Map Descriptor and data will be stored in the DA_Current Data Array in the following fashion if parameter HCSR_Data_Stages >1:

- New data is stored as current data.
- If data is changed, last current data is moved to 2nd Stage, 2nd Stage data is moved to 3rd stage and 3rd stage data is lost.

Appendix A.1.1. Data_Type is Ana_Input or not specified

Data Array Offset	Description
Current Data	
0 - 18	Data bits (0 - 1)
19	Receipt ok (0), Disturbance (1)
20	From Loop (0), On Loop (1)
21	Loop Number (1,2,3,4) as integer
2nd (Previous) Data	
24 – 42	Data bits (0 – 1)
43	Receipt ok (0), Disturbance (1)
44	From Loop (0), On Loop (1)
45	Loop Number (1,2,3,4) as integer
3rd (Prior to previous) Data	
48 – 66	Data bits (0 – 1)
67	Receipt ok (0), Disturbance (1)
68	From Loop (0), On Loop (1)
69	Loop Number (1,2,3,4) as integer

Appendix A.1.2. Data_Type is Digi_Input

Data Array Offset (Say 0)	Description
Current Data	
0 - 18	Data bits (0 - 1)
19	Receipt ok (0), Disturbance (1)
20	From Loop (0), On Loop (1)
21,22,23	3 bits for Loop Number Offsets 21 22 23 1 0 0 –Loop 1 0 1 0 –Loop 2 1 1 0 –Loop 3 0 0 1 –Loop 4
2nd (Previous) Data	
24 – 42	Data bits (0 – 1)
43	Receipt ok (0), Disturbance (1)
44	From Loop (0), On Loop (1)
45	Loop Number (1,2,3,4) in 3 bits as above
3rd (Prior to previous) Data	
48 – 66	Data bits (0 – 1)
67	Receipt ok (0), Disturbance (1)
68	From Loop (0), On Loop (1)
69	Loop Number (1,2,3,4) in 3 bits as above

The same data will be stored in a Loop based Data Array which will then have the last known data for each loop

Data Array Offset	Description
Current Data	
0 - 18	Data bits (0 - 1)
19	Receipt ok (0), Disturbance (1)
20	From Loop (0), On Loop (1)
2nd (Previous) Data	
21 - 39	Data bits (0 - 1)
40	Receipt ok (0), Disturbance (1)
41	From Loop (0), On Loop (1)
3rd (Prior to previous) Data	
42 - 60	Data bits (0 - 1)
61	Receipt ok (0), Disturbance (1)
62	From Loop (0), On Loop (1)

Appendix A.2. 2. How Server expects Data:

The Server expects data in the following format in order to constitute its response to HCS-R Clients.

Appendix A.2.1. Data_Type is Ana_Input or not specified

Data Array Offset	Description
0 - 18	Data bits (0 - 1)
19	Receipt ok (0), Disturbance (1)
20	From Loop (0), On Loop (1)
21	Loop Number (0,1,2,3)

Appendix A.2.2. Data_Type is Digi_Input

Data Array Offset (say 0)	Description
0 - 18	Data bits (0 - 1)
19	Receipt ok (0), Disturbance (1)
20	From Loop (0), On Loop (1)
21,22	2 bits for Loop Number Offsets
	21 22
	0 0 -Loop 1
	1 0 -Loop 2
	0 1 -Loop 3
	1 1 -Loop 4

THIS PAGE INTENTIONALLY LEFT BLANK